Ann Oper Res (2007) 155: 311–338
DOI 10.1007/s10479-007-0216-y

Metaheuristics approach to the aircrew rostering problem

Panta Lučić · Dušan Teodorović

Published online: 4 August 2007

Abstract The solution of the aircrew-scheduling problem is represented by a set of rotations developed from a given set of flight segments. Once the set of rotations to be made by aircrew members has been determined, the air carrier must solve the aircrew rostering problem that entails the monthly assignment of aircrew members to planned rotations. This paper attempts to solve the aircrew rostering problem, thus constructing personalized monthly schedules using Simulated Annealing, Genetic Algorithms, and Tabu Search techniques. The developed models are tested on numerical examples that consist of constructing schedules for pilots. Dimensions of the considered examples are characteristic of small and medium-sized airlines.

Keywords Airlines · Aircrew rostering · Simulated annealing · Tabu search · Genetic algorithms

1 Introduction

The airline industry is about rapid transportation of people and goods and is highly competitive. The success of an airline company highly depends on how it has designed the flight schedule it uses. When designing an airline schedule, consideration must be given to transportation demand, the air carrier's economic interests and various operational constraints.

P. Lučić
CSSI, Inc., 400 Virginia Avenue SW, Suite 710, Washington, DC 20024, USA
e-mail: plucic@cssiinc.com

D. Teodorović (✉)
Department of Civil and Environmental Engineering, Virginia Polytechnic Institute and State University, Blacksburg, USA
e-mail: duteodor@vt.edu

D. Teodorović
Faculty of Transport and Traffic Engineering, University of Belgrade, Belgrade, Serbia

Passengers, on one hand, are interested in the greatest possible flight frequency, departure times that are adapted to their desires, a high probability of finding a vacant seat on a particular flight, short waits to continue traveling at hubs, few cancelled and late flights, etc. On the other hand, the carrier's interests are in a high load factor, high annual aircraft utilization, low operating costs, and high profit. The air carriers are interested in airline schedules that result in a well "filled" airplane and good utilization of existing transportation capacities. Certain passenger requests regarding the airline schedule inevitably conflict with the carrier's requests. The airline schedule design must find the best possible way to reconcile carrier and passengers' conflicting requirements.

When creating a schedule, various operational requirements should be carefully studied and taken into account. The following are the most important factors: (a) Airline Maintenance Department requirements (frequency, location, and duration of aircraft service are prescribed within the aircraft maintenance schedule); (b) Aircraft and aircrew availability (numbers of various aircraft types, gate availability, pilots, flight attendants, ground service personnel, and customer service personnel); (c) Crew working hours regulations and training requirements.

When developing the airline schedule, requirements from the aircrew-planning department should also be studied in detail because aircrews have legal, airline, and union rules that precisely regulate aircrew working hours and working duties. Similarly, it is necessary to carefully plan manpower (ground service personnel that handle aircraft on the ground, and customer service personnel that serve passengers in the terminal) for every station in the network. Due to the fact that a flight aircrew primarily has licenses for one specific aircraft type, care must be taken that there are enough pilots, copilots and in some instances flight engineers for every aircraft type used. Safety reasons, relevant laws, and company regulations precisely define the aircrew's working hours, maximum allowed daily take-offs and landing, needed rest, etc. Thus, when developing the airline schedule, attention must also be given to time windows during the day and airports wherein aircrew replacements will be made, duration of time aircrew stay at certain airports while waiting to take on a job, the number of times aircrew members travel as normal passengers in order to take a job or to return to the base ("dead-heading crews"), and all other factors related to the work of flight aircrew and the work of flight attendants.

On an average, aircrew costs represent 10–15% of total air carrier costs. Consequently, the problem of scheduling the airline aircrew and assigning them to flights must be given appropriate consideration. The total number of aircrew, their utilization, and total aircrew costs depend to a large extent on the chosen airline schedule alternative. However, although there is a significant interdependence between the aircraft schedule design process and the aircrew scheduling process, these two processes are usually separated in order to facilitate problem solving. In air transportation, as in other fields of transportation, the vehicle schedule is usually designed first and then the crew is scheduled based on that. This approach is (with few exceptions) strictly followed in majority of the literature devoted to airline scheduling problems.

Different operational constraints regarding the aircrews' work must be taken into account when assigning aircrews to specific flights and rotations. The most important constraints are the length of the aircrew's working time, regulated payment for overtime, etc. Solution of the aircrew-scheduling problem is represented by a set of rotations from a given set of flight segments (legs). An aircrew rotation could be defined as a sequence of flight segments on successive days made by an aircrew that leaves from and returns to the same air carrier hub. Once the set of rotations to be made by aircrew members has been determined, the air carrier must solve the aircrew rostering problem. The aircrew rostering problem consists of assigning individual crewmembers to the planned aircrew rotations. Crucial part of the aircrew rostering problem is to construct personalized monthly schedules (rosters). When the aircrew rostering problem has been solved, every aircrew member will be assigned a set of rotations to be made the following month.

Aircrew scheduling and aircrew rostering problems are frequently of large dimensions. They are combinatorial optimization problems by their nature, and belong to the class of NP-hard problems. Various heuristic techniques that can generate solutions of reasonably good quality in an acceptable amount of computer time have been used when solving aircrew scheduling and aircrew rostering problems. Recently, more exact approaches have been developed.

During the last two decades, some powerful metaheuristics have been developed and successfully applied when solving many real-life problems. In this paper, an effort is made to develop models based on local search and population search for good solutions of the aircrew rostering problem. Approaches to the aircrew rostering problem based on Simulated Annealing, Tabu Search, and Genetic Algorithm have been explored. The approaches developed in this paper are tested on problems that consist of constructing schedules for pilots. In the problem of constructing schedules for flight attendants, other constraints must be introduced, including those related to the composition of the aircrew.

Real-life crew rostering problems are often characterized by more than one objective function. Combinatorial nature, big dimensions, and existence of more than one objective function have led the authors of this paper to solve the aircrew rostering problem using various meatheuristic algorithms.

The paper is organized as follows: Sect. 2 is devoted to the Basic Characteristics of the Airline Scheduling Problem. The aircrew rostering problem is described in Sect. 3. In that section, we also give a brief review of the literature devoted to the aircrew rostering problem. Section 4 contains a mathematical formulation of the aircrew rostering problem. The proposed solution to the aircrew rostering problem is given in Sect. 5. Numerical results are provided in Sect. 6. Section 7 presents the concluding remarks.

2 Basic characteristics of the airline scheduling process

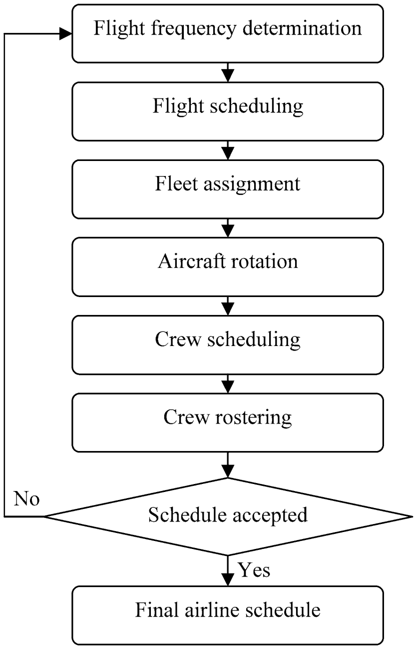
The schedule's design process may start as early as three months or even one year before its execution. This process is graphically shown in Fig. 1.

The airline marketing department provides basic information related to various airline markets to the airline scheduling department. Within the phase of Flight Frequencies Determination (Fig. 1), flight frequencies are established between particular city pairs. Flight scheduling process is aimed to determine "legs" (leg represents non-stop flight between two airports and is described by the origin, destination, departure time and flight time).

Airlines usually operate with various fleets each composed of a particular type of aircraft. In the Fleet Assignment phase, each leg defined in the Flight Scheduling phase is assigned to a specific fleet. In this planning phase, the airline tries to answer the following question: Which aircraft type should fly each leg? When doing this assignment, an airline must take care about the available number of aircraft of each type, as well as about the cost of assigning certain aircraft type to a particular leg.

An aircraft rotation (aircraft route) could be defined as a sequence of legs flown by the aircraft of the same type (aircraft that belong to the same fleet). Rotation duration could be one, two, three, or several days. Usually, these aircraft rotations begin and end in the same air carrier base. In cases when this is not required, analysts face an additional problem of balancing the airline fleet across the different carrier bases.

Fig. 1 Airline schedule planning process



Fleet assignment models assign aircraft types to each of the planned flight legs. These models indicate for each flight leg only aircraft type, and do not assign specific aircraft from the fleet ("tail numbers") to flight legs. Aircraft routing models create aircraft routes (aircraft rotations) by assigning specific aircraft ("tail numbers") to one or more flight legs. Thus, the aircraft routing models provide more specific information and help in determining the needed number of aircraft and their routes during a fixed time period (usually one week). When creating a set of aircraft rotations, one must consider the airline maintenance department's requirements. In the next step, since aircraft have been routed, an analysis could be performed of the schedule's flexibility, i.e., whether certain changes in the schedule might decrease the number of aircraft needed.

Once the airline schedule has been designed (fight scheduling, fleet assignment, and aircraft rotation), the air carriers' next tasks involve scheduling aircrews and assigning them to planned flights. Airlines, as a rule, separate aircrew planning into: (a) crew scheduling (crew pairing) and (b) crew rostering (crew assignment) phase.

In the *crew-scheduling* phase, aircrew rotations are generated out of the flight legs. When generating aircrew rotations (pairings), planners must assign crew to every flight leg (Teodorović 1988; Hoffman and Padberg 1993; Vance et al. 1997, Klabjan et al. 2001, 2002). An aircrew rotation is a sequence of legs on consecutive days made by a crew that begins and ends in the same air carrier's base.

Once the aircrew-scheduling problem is solved, air carriers are faced with the problem of *crew rostering*. Within the crew rostering phase, the generated aircrew rotations (pairings) together with some other crew duties (ground duties, reserve duties, and off-duty blocks) are sequenced to rosters. These rosters are then assigned to individual aircrew members. In other words, the aircrew rostering problem includes the construction of personalized monthly schedules (rosters). There are different types of aircrew rostering problems: those where employees express preferences for some tasks, those where preferences must consider seniority order, those where preferences must be satisfied globally, and those where no preferences are expressed (which is the problem addressed in this paper). When the aircrew rostering problem has been solved, each crewmember will be assigned rotations to be made during the following month.

3 Basic characteristics of the aircrew rostering problem

The aircrew rostering problem consists of assigning individual crewmembers to the planned aircrew rotations. The working duties each aircrew member will be assigned must be determined for every day of the following month. During this assignment, individual crewmembers are assigned to the planned aircrew rotations (Fig. 2). A number of airlines attempt to minimize total crew costs, some try to maximize crew utilization and minimize the number of reserve crewmembers, etc. When the crew rostering problem has been solved, every aircrew member is assigned a set of rotations to be made the following month. When assigning aircrew members to rotations, consideration is given to planned vacations, sick leave, days planned for schooling and training, requested free days, days when medical examinations are held, etc.

The aircrew rostering problem was studied among others, by Agard (1970), Nicoletti (1975), Antosik (1978), Buhr (1978), Moore et al. (1978), Tingley (1979), Marchettini (1980), Giafierri et al. (1982), Glanert (1984), Sarra (1988), Byrne (1988) and Ryan (1992, 2000), Anantaram et al. (1993), Gamache and Soumis (1993), Day and Ryan (1997), Teodorović and Lučić (1998), Lučić and Teodorović (1999), El Moudani and Mora-Camino (2000), and El Moudani et al. (2001).

The usual approach is to treat the aircrew rostering problem as a zero-one integer-programming problem. Buhr (1978) proposed minimizing the difference between the average monthly flight time per crewmember and the monthly flight time of individual crewmembers. Buhr (1978) also indicated the possibility of minimizing the discrepancy between the average number of days on duty during the month and the number of days individual crewmembers are on duty. Antosik (1978) minimized overtime work expenses and the number of pilots who fly less than the guaranteed amount. Gamache et al. (1994) strove to minimize the total duration of uncovered rotations. Some aircrew members have a higher priority than others. Many airlines strictly follow the "seniority" principle. Also, some tasks have a higher priority (some working duties are more "popular") than others. In some cases (Marchettini 1980; Glanert 1984), the aircrew rostering problem is solved by assigning higher priority tasks to higher priority aircrew members. Caprara et al. (1998) proposed a general model for airline/railway applications. The authors also proposed a heuristic algorithm for its solution. The proposed heuristic that constructs one roster at a time was tested in a case of Italian Railways. David et al. (2001) proposed an enhanced rostering model that allows "downgrading as a means of solving tight crew problems". The suggested approach represents a modification of the branch-and-bound technique. The authors tested the proposed model and the developed SWIFTROSTER algorithm in the case of medium-sized European airline (1300 crew members and six different aircraft types). Various problems (ranging from small instances with only 14 pilots and 64 pairings up to 779 crew members and 1711 pairings) were solved using the proposed approach. The obtained CPU-times allow consideration of the real-world applications. Yan et al. (2002) proposed a few crew rostering models. The proposed models are formulated as integer linear programs. The authors tested the proposed approach using the data related to the international operations of China Airlines. The obtained results were very good. Kohl and Karisch (2004)

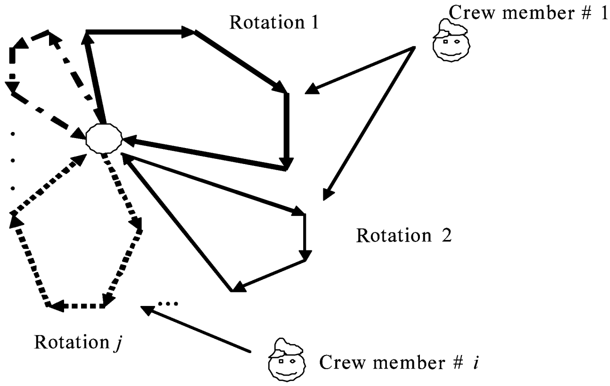


Fig. 2 Assigning individual crewmembers to the planned aircrew rotations

presented a broad description of real-world airline crew rostering problems. They also analyzed various mathematical models used by the airline industry. The authors considered many practical aspects, exposed the complexity of real-world aircrew rostering problems, and presented the solution methods in use in commercial crew rostering systems (methods in use in British Airways, KLM, Iberia, Alitalia, and Scandinavian Airlines (SAS)). When considering future research, Kohl and Karisch (2004) suggested integration of the crew pairing and crew rostering into one planning problem. In other words, they suggested formation of rosters out of legs. They also indicated the crew recovery problem as one of the most important problems to be studied in future research.

The most valuable sources of information related to the previous research in the area of crew rostering are papers of Ernst et al. (2004a, 2004b). Ernst et al. (2004a) made a detailed review of applications, methods and models for staff scheduling and rostering. The authors analyzed 200 references related to various application areas, and proposed a classification scheme for describing rostering problems. Ernst et al. (2004b) produced an annotated bibliography of personnel scheduling and rostering. Their annotated bibliography is composed of approximately 700 references related to the algorithms for generating rosters and personnel schedules, workforce planning and staffing requirements. The authors classified all papers according to the type of problem studied, the application areas covered and the methods used. They also provided a short summary of every considered paper. Freling et al. (2004) developed a decision support system for airline and railway crew planning. They used branch-and-price approach for crew scheduling and crew rostering. The authors explained various implementation issues and performed detailed computational experiments. They compared the results of the approach when crew scheduling is performed before crew rostering, with the results of the case when scheduling and rostering are solved in an integrated way. Cappanera and Gallo (2004) formulated the airline crew rostering problem as a 0–1 *multicommodity flow problem.* In theirs formulation, each crewmember corresponds to a commodity. Computational experiments were performed using a commercial integer-programming solver (CPLEX).

Because of safety reasons, there is the inclination in airline industry to adapt the pilot as much as possible to a certain aircraft type. In other words, the pilots of most world airlines usually fly only one aircraft type. It is really rare for a pilot to fly several types of aircraft during the month. This means that in most cases monthly schedules are first made for the pilots of one aircraft type, then for the pilots of another type of aircraft, etc. The situation with the flight attendants is quite opposite. Flight attendants frequently have licenses to work on all types of aircraft in the airline's fleet.

In this paper, we consider the situation in which pilots are separated into groups according to the aircraft type for which they have a certificate. The aircrew rostering problem is solved within the framework of pilot groups that fly on the same type of aircraft.

4 The aircrew rostering problem: mathematical formulation

It is usual practice in airline industry to measure the maximum amount of aircrew workload by the maximum amount of flight time, as well as by the maximum number of take-offs. Law in every country precisely prescribes the maximum amount of aircrew workload. Usually, it is also allowed that every airline can define its own internal constraints that are stricter than those defined by law. There are various aircrew workload constraints. Some of them are related to a single day, while other constraints refer to a longer time period (usually one month). Constraints that refer to a single day (maximum allowed daily flight time, maximum allowed daily number of take-offs) are considered when constructing aircrew rotations and will not be considered in this paper any further.

Let us denote by m the total number of aircrew members to be assigned to work duties during the following month. The total number of rotations to be made the following month is denoted by k.

Let us introduce the following parameters:

$$p_{il} = \begin{cases} 1 & \text{if the } i\text{-th crewmember can spend the } l\text{-th day at work,} \\ 0 & \text{otherwise,} \end{cases} \qquad \begin{matrix} i = 1, 2, \ldots, m, \\ l = 1, 2, \ldots, 30. \end{matrix}$$

When $p_{il} = 0$, then it has been planned in advance for the i-th crewmember to be absent from work on the l-th day (vacation, medical examination, schooling and training, etc.) The total number of possible working days P of all m crewmembers during the following month equals:

$$P = \sum_{i=1}^{m} \sum_{l=1}^{30} p_{il}. \tag{1}$$

We denote by d_j the "length" of the j-th rotation expressed as flight time. Total "length" D of all k rotations to be made the following month is:

$$D = \sum_{j=1}^{k} d_j. \tag{2}$$

We denote also by a, the ideal average daily flight time of each of the m crewmembers. This number equals:

$$a = \frac{D}{P} = \frac{\sum_{j=1}^{k} d_j}{\sum_{i=1}^{m} \sum_{l=1}^{30} p_{il}}. \tag{3}$$

The best situation upon solving the aircrew rostering problem would be for each of the m crewmembers to have an average daily flight time equal to a. Let us denote by a_i^* the ideal monthly flight time of the i-th crewmember. This is:

$$a_i^* = a \sum_{l=1}^{30} p_{il}. \tag{4}$$

As we can see, when calculating the ideal monthly flight time for each crewmember, the total number of days in the month that the crewmember can be assigned to a work duty must be taken into consideration. Value a_i^* clearly differs from pilot to pilot.

Let us introduce into the discussion, binary variable x_{ij} that is defined as follows:

$$x_{ij} = \begin{cases} 1 & \text{if the } i\text{-th crewmember is assigned to the } j\text{-th rotation,} \\ 0 & \text{otherwise.} \end{cases}$$

We denote by $f_1(x)$ the average relative deviation (per crewmember) of real monthly flight time from the ideal raised to the r-th power. Then:

$$f_1(x) = \frac{1}{m} \sum_{i=1}^{m} \left| \frac{\sum_{j=1}^{k} d_j x_{ij} - a_i^*}{a_i^*} \right|^r. \tag{5}$$

Models that minimize deviations are always vulnerable to solutions that include one large deviation. Although mathematically feasible, in reality such a solution might be considered unfair or unreasonable. The goal of introducing degree r ($r > 1$) is to eliminate significant deviations between real flight time and ideal flight time that can appear for a certain number of crewmembers when $r = 1$. Value r is a constant that is given in advance. It is clear that the smaller the value of $f_1(x)$, the closer the real monthly flight time of most crewmembers is to the ideal possible monthly flight time. Value $f_1(x)$ can serve as one of the objective functions when solving the aircrew rostering problem. Of course, instead of minimizing the average monthly deviation (per crewmember), it would be possible to minimize the maximum deviation per crewmember, etc.

The relatively equal workload per crewmember can also be understood as an equal number of weekend days spent outside the home, an equal number of departures before 7:00 a.m., an equal number of foreign per diem allowances during the month, etc. In some companies aircrews are much better paid when they fly routes outside domestic airspace. It was shown that in the company used as an illustrative example in this paper, equaling the number of "special" per diem allowances per crewmember was extremely important. The special per diem included a special monetary compensation received by the pilot for every workday spent on duty on an international route.

Let us further denote by b the "ideal" share of foreign per diem allowances that each crewmember should receive for each day of work availability during the month. Value b equals:

$$b = \frac{\sum_{j=1}^{k} c_j}{\sum_{i=1}^{m} \sum_{l=1}^{30} p_{il}}, \tag{6}$$

where c_j is the total number of foreign per diem allowances provided in the j-th rotation.

Let further b_i^* be the ideal monthly number of foreign per diem allowances of the i-th crewmember. Then b_i^* equals:

$$b_i^* = \left[b \sum_{l=1}^{30} p_{il} \right], \tag{7}$$

where $[y]$ denotes real number y rounded to the nearest integer.

We denote by $f_2(x)$ the average absolute deviation (per crewmember) between real and ideal number of foreign per diem allowances during the month raised to the r-th power.

Then:

$$f_2(x) = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{\sum_{j=1}^{k} c_j x_{ij} - b_i^*}{b_i^*}\right|^r. \tag{8}$$

Let $\overline{v_i}$ be the number of weekend days i-th crewmember can spend on duty. Then:

$$\overline{v_i} = \sum_{l=0}^{\lfloor\frac{30-\eta}{7}\rfloor} p_{i,\eta+7l} + \begin{cases} \sum_{l=0}^{4} p_{i,1+7l}, & \text{for } \eta = 7, \\ \sum_{l=0}^{\lfloor\frac{30-\eta-1}{7}\rfloor} p_{i,\eta+1+7l}, & \text{for } \eta = 1,\dots,6, \end{cases} \tag{9}$$

where η is an ordinary number of the first day of the weekend during the month and $\lfloor y\rfloor$ is real number y rounded down to the closest integer.

Let us denote by v the "ideal" share of workdays over the weekend that each crewmember should receive for each weekend day of work availability during the month. Value v equals:

$$v = \frac{\sum_{j=1}^{k} u_j}{\sum_{i=1}^{m} \overline{v_i}}, \tag{10}$$

where u_j is the total number of weekend days covered in the j-th rotation.

Further, let v_i^* be the ideal monthly number of weekend days spend on duty of the i-th crewmember. Then v_i^* equals:

$$v_i^* = [v \quad \overline{v_i}]. \tag{11}$$

We denote by $f_3(x)$ the average absolute deviation (per crewmember) between real and ideal number of weekend days on duty during the month raised to the r-th power. Then:

$$f_3(x) = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{\sum_{j=1}^{k} u_j x_{ij} - v_i^*}{v_i^*}\right|^r. \tag{12}$$

The problem we are considering in this paper is concerned with minimization in the multi-objective sense of three functions $f_1(x)$, $f_2(x)$ and $f_3(x)$ over finite set X. For any of the pair of points$(x, y) \in X \times Y$, one of the following alternatives must be satisfied: (1) $x \prec y$ (x is preferred to y), (2) $x \succ y$ (y is preferred to x), (3) $x \sim y$ (x and y are indifferent) or (4) $x?y$ (no preference can be stated between x and y).

To define preference structure in this paper, scalar ordering is used as follows:

If $F(f(x)) < F(f(y))$ then x is preferred to y; if $F(f(x)) = F(f(y))$ then the two solutions are indifferent. The $F : R^3 \to R$ is simple scalar function. Let:

$$F : F(f) = \sum_{i=1}^{3} w_i f_i, \tag{13}$$

where w_i are nonnegative weights of objective functions satisfying $\sum_{i=1}^{3} w_i = 1$.

Typical one-month constraints of the aircrew rostering problem could be outlined as follows:

(a) The total flight time accumulated during the entire month must not exceed 85 hours.

$$\sum_{j=1}^{k} d_j x_{ij} \le 85, \quad i = 1, 2, \dots, m. \tag{14}$$

(b) The total number of takeoffs per pilot per month must not exceed 90.

$$\sum_{j=1}^{k} f_j x_{ij} \leq 90, \quad i = 1, 2, \ldots, m, \tag{15}$$

where f_j is the number of takeoffs contained in the j-th rotation.

(c) The total monthly number of working hours must not exceed 160 h per pilot. If we denote by D_j the number of working hours needed to complete the j-th rotation, then:

$$\sum_{j=1}^{k} D_j x_{ij} \leq 160, \quad i = 1, 2, \ldots, m. \tag{16}$$

(d) Every pilot must have a free day no later than the fifth consecutive working day. Let us introduce the following parameters into the discussion:

$$q_{jl} = \begin{cases} 1 & \text{if the } j\text{-th rotation starts on the } l\text{-th day of the month,} \\ 0 & \text{otherwise,} \end{cases} \quad \begin{array}{l} j = 1, 2, \ldots, k, \\ l = 1, 2, \ldots, 30. \end{array}$$

If we denote by t_j, the total number of days required for an aircraft crew to make the j-th rotation, then the following condition must be fulfilled when assigning crews to rotations:

$$\sum_{j=1}^{k} t_j x_{ij} \sum_{l=p}^{p+5} q_{jl} \leq 5, \quad i = 1, 2, \ldots, m, \ p = 1, 2, \ldots, 25. \tag{17}$$

The rest period pattern: "Pilots should take a rest for one day, after five successive working days at the pilot's home base" is quite simple compared with the other patterns that could be found in the airline industry. In some airline companies, employees must have either 3 periods of 3 days off, or 3 periods of 2 days and a period of 4 days, or 2 periods of 4 days, etc. When different choices of patterns must be taken, then constraints (17) become much more complicated.

Remaining constraints are as follows:

While carrying out his rotation, a pilot may not be given a free day.

$$\sum_{j=1}^{k} x_{ij} = \sum_{j=1}^{k} x_{ij} \sum_{l=1}^{30} q_{jl} \prod_{s=l}^{l+t_j-1} p_{is}, \quad i = 1, 2, \ldots, m. \tag{18}$$

These constraints simply prohibit assignment of rotations to aircrews during their scheduled leave. To maintain the solution feasibility, rotations assigned to a crewmember should not overlap in time with his/her days on scheduled leave.

One rotation may be given to only one pilot.

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j = 1, 2, \ldots, k. \tag{19}$$

The rotations assigned to a pilot must not overlap in time. Let us denote by T_j the moment in time (day and time) when a pilot starts the j-th rotation. We also denote by T'_j the moment in time when the pilot is ready to take a new rotation after completing the j-th rotation. The time interval in which the pilot is busy with the j-th rotation is $[T_j, T_j')$. Rotations j and p overlap if the cross section of their time intervals, thus defined, is not an empty set.

Let us introduce into the discussion the following parameters:

$$\rho_{rs} = \begin{cases} 1 & \text{if rotation } r \text{ overlaps with rotation } s, \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{ij} \sum_{s=1}^{k} \rho_{js} x_{is}(s-j) = 0, \quad i = 1, 2, \ldots, m, \; j = 1, 2, \ldots, k. \tag{20}$$

When solving the crew rostering problem in this paper, emphasis was put on achieving the smallest possible values for $f_1(x)$, $f_2(x)$ and $f_3(x)$, while at the same time satisfying all existing constraints.

5 Solving aircrew rostering problem by metaheuristics

In this paper we try to use some of the metaheuristic methods to find not the optimal solution but a "satisfactory solution" to the aircrew rostering problem. A "satisfactory solution" in this paper is understood to be a solution that enables all aircrew members to have "an approximately equal work load." We use Simulated Annealing (Metropolis et al. 1953; Kirkpatrick et al. 1983; Cherny 1985), Tabu Search (Glover 1986, 1989, 1990a, 1990b; Glover and Laguna 1993), and Genetic Algorithm (Holland 1975; Goldberg 1989) techniques to solve the aircrew rostering problem.

Many heuristic techniques that have been developed are capable of solving only specific problems. On the other hand, metaheuristics can be defined as general optimization techniques capable of solving different optimization problems. The developed metaheuristics are based on local search or on population search. Local search based metaheuristics (Simulated Annealing, Tabu Search, etc.) are characterized by an investigation of the solution space in the neighborhood of the current solution. Each step in these metaheuristics represents a move from the current solution to another potentially good solution in the current solution's neighborhood. Population search based metaheuristics, however, simultaneously evaluate a population of solutions. These solutions are modified and the new generation of solutions is generated. Each new generation of solutions is expected to be "better" than the previous one.

In order to apply any of the known metaheuristics (based on either the local search or the population search), we need a method that will provide the initial solution (frequently it is also required to be a feasible solution). In this paper, the initial feasible solution(s) are generated using the "pilot-by-pilot" greedy heuristic.

5.1 The "pilot-by-pilot" aircrew rostering heuristic

The "Pilot-by-pilot" method is a simple greedy heuristic that sequentially defines the monthly assignment for the first crewmember, then for the second one, etc. This method is composed of the following algorithmic steps:

Algorithm 1 ("Pilot-by-pilot" algorithm)

Step 1. Create a list of aircrew members.
Step 2. Select an aircrew member from the list of aircrew members.
Step 3. Assign the aircrew member to specific rotations so that all operational constraints are satisfied.
Step 4. Take away the aircrew member and all rotations assigned to him/her from further consideration.
Step 5. Verify whether stopping criteria is fulfilled (all crewmembers are assigned to the rotations, or all the rotations are assigned to the crewmembers). If affirmative and if solution is feasible the algorithm is finished. Otherwise go to Step 1. If stopping criteria is not fulfilled, go to Step 2.

The list of crewmembers (Step 1) could be created using "seniority" or some other principle. In case a feasible solution could not be found, the order of crewmembers on the list could be changed randomly.

The "Pilot-by-pilot" heuristic algorithm is fast. On the other hand, this algorithm does not provide assurance about the quality of the obtained solution or about its feasibility. If there are some rotations not assigned to the crewmembers, then the obtained solution is infeasible. Appropriate sorting of crewmembers can help us to reach a feasible solution. For example, crewmembers may be sorted in ascending order based on their work availability. In this case, the most constrained crewmembers would be assigned to the rotations at the very beginning.

The following simple tests show whether an instance of the aircrew rostering problem has a feasible solution:

(1) For each day during the month the number of available crew members exceeds the number of rotations,
(2) An average number of flight hours, take-offs and work hours per aircrew member should not exceed 85, 90 and 160 respectively.

Finding a feasible solution of the aircrew rostering problem could be a difficult and challenging task in some instances. In some extreme cases, the following problem could appear: For a given set of constraints assign the rotations to the aircrew members in such a way to minimize the total number of rotations not being covered by the aircrews. Instances of the aircrew rostering problem having tight solution space are out of the scope of this paper.

5.2 Solving aircrew rostering problem by simulated annealing

The algorithm for solving aircrew rostering problem by Simulated Annealing developed in this paper is a two-step heuristic. We first generate an initial feasible solution by "pilot-by-pilot" method, and then we try to improve the initial solution using the simulated annealing technique.

Simulated Annealing (SA) is based on an analogy between the way in which a metal cools into a minimum energy crystalline structure (the annealing process) and the random search for a minimum state of a more general system. Kirkpatrick et al. (1983) and independently Cherny (1985) have developed an optimization method based on physical process described in Metropolis et al. (1953). SA is a random search technique which avoids being trapped at local minima by accepting not only changes (solutions) that improve objective

Table 1 Representation of the pilots' monthly assignment to individual rotations

Pilots	**17**	18	32	...	...	48	**25**	66	...	...	44	26
Rotations	**1**	2	3	...	...	55	**56**	57	...	...	$k-1$	k

Table 2 Exchanging rotations among pilots

Pilots	**17**	18	32	...	...	48	**25**	66	...	...	44	26
Rotations	**1**	2	3	...	...	55	**56**	57	...	...	$k-1$	k

function (f) but also some changes that worsen it. The latter are accepted with a probability:

$$P = e^{-\frac{\Delta f}{T}}, \tag{21}$$

where Δf is the increase in f gained while changing a state from previous to new one and T is a control parameter, which by analogy with the Metropolis et al. (1953) application is known as the "temperature."

In implementation of the simulated annealing algorithm, there is always an old solution and a new solution (obtained by random perturbation of the old solution) whose "quality" is being examined. It has been suggested by Eglese (1990) that throughout the completing of the simulated annealing algorithm, the best solution obtained thus far should always be kept. We have followed that principle, so that we always judge the existing (old) solution, the new solution being examined, and the best solution ever obtained.

In this paper, we make perturbations at random in order to give all rotations the opportunity to replace the pilots that fly them. We prescribe that one perturbation consists of a certain number of moves. We also denote by n the number of moves that make up each perturbation (the number n is subjectively estimated by the analyst). Let us study the change in the solution caused by one move (Table 1). For example, the first rotation will be made by pilot no. 17, the second by pilot no. 18, the third rotation by pilot no. 32, etc. Let us note rotation 1 flown by pilot no. 17. We randomly select a new pilot to fly the rotation 1 instead of pilot 17. Let this be pilot no. 25.

The assumption is that pilot 25 can be assigned to fly rotation 1 instead of pilot 17 (we conclude this after checking all operational constraints). Now, we can replace pilot 17 with pilot 25 at the position corresponding to rotation 1 (Table 2).

Pilot 17 has reduced flight time, simultaneously pilot 25 has increased flight time. In this paper, we call this substitution of pilot 17 by pilot 25 the *"subtracting-adding"* scheme since one pilot loses a rotation and another one gains a rotation. When we apply this scheme, in certain cases, the new rotation assigned to a pilot can significantly raise his/her total amount of flight time. Obviously, such unexpected changes can obstruct the search process, especially in the final search phases. In order to stay away from this, in this paper, the *"exchanging"* scheme is applied in some cases where two pilots trade one rotation each. Under the *"exchanging"* scheme, pilot 25 would be assigned rotation 1 that was to be flown by pilot 17, and at the same time pilot 17 would be assigned *some* rotation that had been previously planned for pilot 25. An algorithm has been developed to make the moves. The algorithm that uses both the *"subtracting-adding"* scheme and the *"exchanging"* schemes consists of the following steps.

Table 3 The old, and the new monthly assignment of pilots to rotations

Pilots	**17**	18	32	...	...	48	**25**	66	...	...	44	26
Rotations	**1**	2	3	...	...	55	**56**	57	...	...	$k-1$	k

Pilots	**25**	18	32	...	...	48	**17**	66	...	...	44	26
Rotations	**1**	2	3	...	...	55	**56**	57	...	...	$k-1$	k

Algorithm 2 (Algorithm for exchanging rotations among pilots)

Step 1. Consider a rotation that is in order to have its pilot reinstated. Choose randomly a pilot-candidate to perform the rotation. Attempt to assign the rotation to the pilot-candidate using the *"subtracting-adding"* scheme. If the rotation cannot be assigned to this pilot due to operational constraints, randomly choose a second pilot-candidate, then the third pilot candidate, etc. If after z attempts, no pilot-candidate has been found who can accept the rotation, there will be no replacement of the pilot flying that rotation. Go to Step 5. When the *"subtracting-adding"* scheme is used and the pilot can be replaced on the rotation in question, check the following inequality related to the whether the pilot who has had the rotation subtracted (after subtraction):

$$\min\left\{\frac{\sum_{j=1}^{k} d_j x_{ij} - a_i^*}{a_i^*}, \frac{\sum_{j=1}^{k} c_j x_{ij} - b_i^*}{b_i^*}, \frac{\sum_{j=1}^{k} u_j x_{ij} - v_i^*}{v_i^*}\right\} \geq 0. \qquad (22)$$

If the inequality is satisfied, go to Step 5 (in this case the move is completed using the *"subtracting-adding"* scheme). In the opposite case, go to Step 2.

Step 2. Sort all the rotations formerly assigned to the pilot-candidate (not including the rotation in question) in descending order according to the rotation "length." The rotation "length" units depend on the most negative value in relation (22). (For example, if after giving up rotation, a crewmember turns out to be short of special per diem allowances only, the sort of rotations would be done based on the rotation "length" expressed in special per diem allowances.) Go to Step 3.

Step 3. Attempt to assign the first rotation from the rotation list formed in Step 2 to the pilot who has had a rotation subtracted. Should this not be feasible due to operational constraints, attempt to assign the second rotation from the list, then the third, etc. Should it not be feasible to assign the pilot who has given a rotation any of the rotations on the list, go to Step 5 (in this case the move is also completed using the *"subtracting-adding"* scheme). If it is recognized that the pilot who has given a rotation can be assigned a rotation from the list, go to Step 4.

Step 4. Assign the pilot who has given a rotation the rotation from Step 3 for which it has been recognized that its assignment does not violate operational constraints (In this case, the move is completed using the *"exchanging"* scheme.) Go to Step 5.

Step 5. Stop.

Let us assume that after using the above algorithm, pilot 17 has been assigned rotation 56 that used to belong to pilot 25 and pilot 25 has been assigned rotation 1 that used to belong to pilot 17 (*"exchanging"* scheme is applied). The old feasible solution and the new feasible solution that is obtained after the move are shown in Table 3.

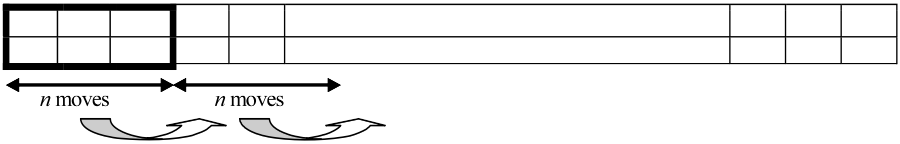


Fig. 3 Sequence of perturbations

Moves are performed one at a time. In other words, after completing one move, the next one is performed within the perturbation (traveling from "left to right" in a direction of increasing rotation numbers). The operational constraints are checked whenever a move is made. These constraints are checked for every pilot alternate so that after the perturbation (n moves) the solution obtained is feasible. One should perceive a perturbation as an attempt to alter the pilots assigned to n rotations. After performing n moves, when the perturbation is completed, it could happen that not a single rotation will change pilots, or that only one rotation will change pilots, or that two rotations will change pilots, ..., or that $2n$ rotations will have new pilots (Fig. 3).

After completing one perturbation, the next one is performed (moving from "left to right" in a direction of increasing rotation numbers).

An *equilibrium* is defined as a set of solutions whose objective function values are not systematically decreasing. Crucial to the understanding of equilibrium conditions is the concept of an *epoch* (Skicsim and Golden (1983). The epoch represents the interval between equilibrium testing. In our case, an epoch consists of certain number of attempted perturbations. We specify the number of attempted perturbations $a\ priori$. There are in total k rotations to be flown. In this paper, we define the epoch in the following way: One epoch consists of $\lceil \frac{k}{n} \rceil$ perturbations, where $\lceil y \rceil$ is the value of real number y rounded up to the nearest integer value. After completing the perturbation, the obtained solution is saved and tested for an equilibrium. The test consists of comparing the objective function value from the most recent epoch with the objective function values from all previous epochs at a specified temperature.

Let us denote by:

f_r—the objective function value obtained at the end of the r-th epoch,
f_s— the objective function value obtained based on the s-th epoch, where $0 < s < r$.

If the objective function value from the most recent epoch is satisfactorily close to any previously observed objective function values, the thermal equilibrium is reached, i.e. the thermal equilibrium is achieved if the following is satisfied:

$$|f_r - f_s| \leq \varepsilon \quad \text{for any } s, \tag{23}$$

where ε is a previously assigned constant.

Reaching thermal equilibrium can also be defined by the condition:

$$\frac{|f_r - f_s|}{f_r} \leq \varepsilon \quad \text{for any } s. \tag{24}$$

When thermal equilibrium is achieved, the next temperature is selected and the procedure is repeated.

The expression for the probability of accepting the new solution (21) as well as the relations describing the thermal equilibrium conditions ((23) and (24)) could be used when solving single objective combinatorial optimization problems. Since we are dealing with the multi-objective aircrew rostering problem, it is necessary to define the corresponding expressions. The probability of accepting the new solution (Serafini 1994; Lučić and Teodorović 1998) is:

$$p_{xy}(T) = \min\left\{1, e^{\sum_{i=1}^{3} \frac{w_i(f_i(x)-f_i(y))}{T}}\right\}. \tag{25}$$

Thermal equilibrium is reached when the following condition is satisfied:

$$\sum_{i=1}^{3} w_i |f_{ir} - f_{is}| < \varepsilon \quad \text{for any } s \ (0 < s < r). \tag{26}$$

In some cases, it might happen that condition (26) would be very difficult to satisfy. Therefore, it is assumed in this paper that thermal equilibrium is also reached after α epochs if there has been no improvement of the solution. Value α is the maximum number of search epochs at one temperature.

In order to apply the simulated annealing technique successfully, one should adequately answer the questions related to the number of elements of the solution that should be replaced so that the changes are small, the initial temperature, the law on changing temperature (the cooling schedule), the maximum number of epochs that must be traversed before moving to a new temperature, etc. The answers to these questions are based on experience and intuition and usually include the running of a great number of computer experiments.

This paper uses *geometric cooling schedule*: $T_i = cT_{i-1}, i = 2, 3, \ldots, q$. Where q is the total number of temperatures and c is a real number from the interval [0.8, 0.99].

Geometric cooling schedule is the most frequently used in the literature; however, it is not the only one. More details on cooling schedules and its impact on the performances of the simulated annealing algorithms are presented in Triki et al. (2005).

The aircrew rostering problem has been solved in this paper using the following simulated annealing algorithm:

Algorithm 3 (Simulated annealing algorithm for the aircrew rostering problem)

Step 1. Establish the temperature schedule $TS = (t_1, t_2, \ldots, t_q)$ where q is the number of different temperatures, such that $t_1 > t_2 > \cdots > t_q$.
Step 2. Using the "pilot-by-pilot" method, create the initial feasible solution (apply Algorithm 1). Denote the current temperature by T. Set $i = 1$ and $T = t_1$.
Step 3. Perform a random perturbation of the generated solution (apply Algorithm 2).
Step 4. Calculate the change in the values of objective functions and acceptance probability $P(accept)$ for the new solution (using relation (25)). If $P(accept) = 1$, go to Step 6. Otherwise, go to Step 5.
Step 5. Using uniform distribution, generate a random number$r \in [0, 1]$. If $r < P(accept)$, go to Step 6. If $r > P(accept)$, keep the old solution and go to Step 7.
Step 6. Accept the new solution and remember the new objective function value.
Step 7. If the thermal equilibrium has been reached (condition (26) is satisfied or search has been already performed in the α epoch) at the present temperature T, set $i = i + 1$. If $i > q$, stop (search has been performed at the last scheduled temperature). Otherwise, set $T = t_i$ and go to Step 3. If thermal equilibrium has not been reached, continue with unchanged temperature T to Step 3.

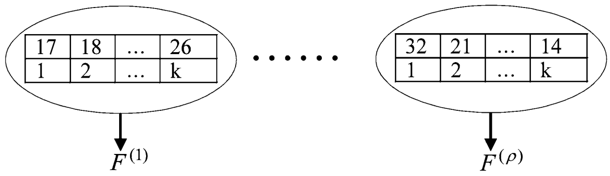


Fig. 4 Different solutions to the crew rostering problem

The proposed algorithm contains two loops. In the outer loop the temperature is altered, while the inner loop determines the number of neighboring perturbations which are to be performed on every temperature.

5.3 Solving aircrew rostering problem by genetic algorithm

Genetic algorithms represent search techniques based on the mechanics of nature selection used in solving complex combinatorial optimization problems. These algorithms were developed by analogy with Darwin's theory of evolution and the basic principle of the "survival of the fittest" (Holland 1975; Goldberg 1989).

In the case of genetic algorithms, as opposed to traditional search techniques, the search is run in parallel from a population of solutions. In the first step, various solutions to the considered maximization (or minimization) problem are generated. In the next step, the evaluation of these solutions that is, the estimation of the objective (cost) function is made.

Some of the "good" solutions yielding a better "fitness" value (objective function value) are further considered. The remaining solutions are eliminated from consideration. The chosen solutions undergo the phases of *reproduction*, *crossover* and *mutation*. After that, a new generation of solutions is produced to be followed by a new one, and so on. Each new generation is expected to be "better" than the previous one. The production of new generations is stopped when a previously specified stopping condition is satisfied. The final solution of the considered problem is the best solution generated during the search. In the case of genetic algorithms an encoded parameter set is used. Most frequently, binary coding is used and the set of decision variables for a given problem is encoded into a bit string (chromosome, individual).

The initial population of solutions $P(0)$ shown in Fig. 4 contains ρ different solutions (strings) to the crew rostering problem. In total k rotations (Fig. 4) should be assigned to the m pilots. In the case of the first solution, the first rotation will be made by pilot # 17, the second by pilot # 18, etc. The rotation # k will be made by pilot # 26. In the case of solution number ρ, the first rotation will be made by pilot # 32, the second by pilot # 21, and the last by pilot # 14. We denote respectively, by $F^{(1)}$ and the $F^{(\rho)}$ values of the objective function (fitness) of string (solution) 1, and string ρ. In general case, let us denote by $F^{(i)}$ the value of the objective function (13) of string i. The probability p_i for string i to be selected for mating is equal to the ratio of $F^{(i)}$ to the sum of all strings' objective function values in the population:

$$p_i = \frac{F^{(i)}}{\sum_{j=1}^{\rho} F^{(j)}}. \tag{27}$$

This type of *reproduction*, that is, selection for mating, represents a proportional selection known as the "roulette wheel selection." (The sections of roulette are in proportion to probabilities p_i.) In addition to the "roulette wheel selection," several other ways of selection for mating have been suggested in the literature. In this paper we use the "roulette wheel selection."

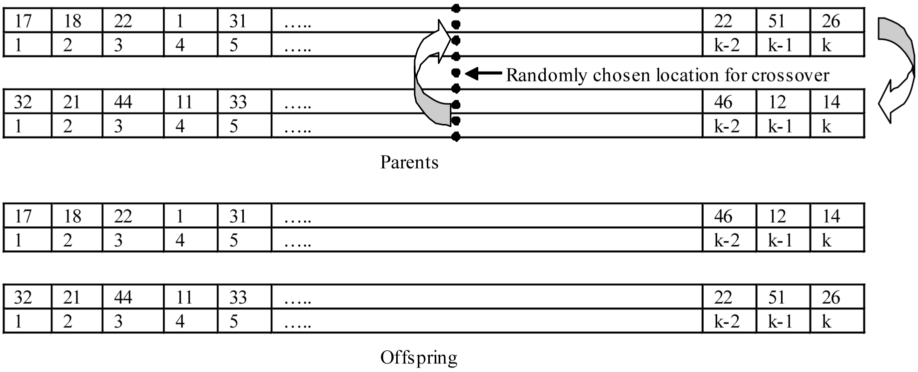


Fig. 5 A single-point crossover operator: **a** two parents, **b** two offspring

In order to generate the next population $P(1)$, we proceed to apply the other two genetic operators to the strings selected for mating. *Crossover* operator is used to combine the genetic material. At the beginning, pairs of strings (parents) are randomly chosen from a set of previously selected strings. Later, for each selected pair the location for crossover is randomly chosen. Each pair of parents creates two offspring (Fig. 5) with crossover probability (p_c). Crossover probability is usually high ranging [0.8, 1].

One or both new solutions (offspring) may be infeasible. Feasibility problem in this article is solved using *"repair"* algorithm. There are many other ways to address the problem of solution feasibility (Coello Coello 2002). There are two potential sources of solution infeasibility: (1) one or more rotations have two crewmembers assigned, or/and (2) one or more rotations have no assigned crewmembers. In order to properly analyze the first problem, we introduce the following attribute of the crewmember i assignment:

$$z_i = w_1 \frac{\sum_{j=1}^{k} d_j x_{ij}}{a_i^*} + w_2 \frac{\sum_{j=1}^{k} c_j x_{ij}}{b_i^*} + w_3 \frac{\sum_{j=1}^{k} u_j x_{ij}}{v_i^*} \quad \text{for } i = 1, \ldots, m. \tag{28}$$

Similarly, for analyzing the second problem we introduce χ_j as the following rotation j attribute:

$$\chi_j = w_1 \Big(d_j / \max_j d_j\Big) + w_2 \Big(c_j / \max_j c_j\Big) + w_3 \Big(u_j / \max_j u_j\Big) \quad \text{for } j = 1, \ldots, k. \tag{29}$$

Introduced quantities z_i (crewmember) and χ_j (rotation) could be used to bring the solution into feasible region by assigning 'not assigned' rotations to crewmembers that need them the most.

Solution repair algorithm used in this paper is as follows:

Algorithm 4 (*"Repair"* algorithm)

Step 1. Select all rotations having two crewmembers assigned. Create the list of these rotations (This list may be created in a random manner). Going through the list of rotations, select the crewmembers i and j having the rotation assigned and do the following: Update values of the quantities z_i and z_j. Remove the rotation from the assignment of crewmember i if $z_i > z_j$. Otherwise, remove the rotation from the assignment of crewmember j.

Step 2. Select rotations with no crewmember assigned. Based on values χ_j, sort them in descending order. Sort all crewmembers into ascending order based on values z_i. Go through the list of crewmembers and assign the first rotation to the first crewmember without violating constraints. Once when rotation is assigned, update the z-attribute values of crewmembers and re-sort their list. Do the same for the second rotation, third rotation, etc.

After completing crossover, the genetic operator *mutation* is used. In the case of binary coding, mutation of a certain number of genes refers to the change in value from 1 to 0 or vice versa. It should be noted that the probability of mutation (p_m) is very small (of order of magnitude 1/1000). The purpose of mutation is to prevent an irretrievable loss of the genetic material at some point along the string. For example, in the overall population, a particularly significant bit of information might be missing (for example, none of the strings have 0 at the seventh location), which can considerably influence the determination of the optimal or near-optimal solution. Without mutation, none of the strings in all future populations could have 0 at the seventh location. Nor could the other two genetic operators help to overcome the given problem. Having generated population P(1) (which has the same number of members as population P(0)), we proceed to use the operators reproduction, crossover, and mutation to generate a sequence of populations P(2), P(3), and so on.

Genetic Algorithm for the aircrew rostering problem consists of the following steps:

Algorithm 5 (Genetic algorithm for the aircrew rostering problem)

Step 1. Randomly create the list of crewmembers (sort them in a random manner). Using the "*pilot by pilot*" method, assign crewmembers to planned rotations. Obtained solution becomes the member of the initial population P(0) that should contain ρ solutions (strings). Randomly create the second list of crewmembers, and by "*pilot by pilot*" method assign crewmembers to planned rotations. The second obtained solution becomes also the member of the initial population P(0). Randomly create the third list of crewmembers, etc. The initial population is finally created after generating ρ solutions (strings). Make an evaluation of the fitness of each string.

Step 2. Considering the fact that the selection probability is proportional to the fitness, select ρ parents from the current population.

Step 3. Randomly select a pair of parents for mating. Create two offspring by exchanging strings with the one-point crossover. Apply the *repair algorithm* if any of the produced offspring is not feasible (Algorithm 4). To each of the created offspring, apply mutation with probability p_m (mutation assumes random selection of two crewmembers where one will provide rotation to the other one using "*subtracting-adding"* scheme which has been previously described). Make an evaluation of the fitness of each offspring. Include in the new population, two best among two parents and two created offspring.

Step 4. Evaluate the fitness of all members in the new population.

Step 5. If the number of generations (populations) is smaller than the maximal predetermined number of generations, go back to Step 2. Otherwise, stop the algorithm. For the final solution choose the best string discovered during the search.

5.4 Solving aircrew rostering problem by tabu search

The tabu search technique is exceptionally useful in solving complex combinatorial optimization problems. Glover (1986) proposed a modern formulation of this technique and Hansen (1986) put forward certain seminal ideas. As Glover and Laguna (1993) note: "Tabu search methods operate under the assumption that a neighborhood can be constructed to identify 'adjacent solutions' that can be reached from any current solution." When permutation problems are concerned, for the purpose of generating a neighborhood, the so-called swaps representing pair-wise exchanges are most often used.

After making a swap, the next solution has a smaller, equal or larger value of the objective function as compared to the previous solution. A complete neighborhood of the current solution includes certain number of different solutions. The tabu search technique uses the concept of the so-called flexible memory. The basic idea is to pronounce the subset of the moves in a neighborhood forbidden. Which moves are to be forbidden (tabu) is decided according to the recency or frequency that certain moves have participated in generating the previous solutions. In other words, by introducing the tabu moves, we try to avoid the swaps made in recent past. Following the tabu status of certain swaps refers to the "recent past." In this sense, it can be said that the tabu search technique is characterized by a "recency-based memory." Glover and Laguna (1993) proposed that the recency-based memory should be followed by the so-called frequency-based memory. This results in a more efficient search that involves the need to both diversify the search and (in certain cases) intensify the search.

Let us assume that the total number of crewmembers equals 5 and that we have already made seventeen iterations. In this paper, swap is defined as an action in which one crewmember provides rotation to the other one according to the "subtracting adding" scheme introduced previously. Let us also assume that the tenure during which a certain swap is in a tabu status equals 2. This means that the same couple of crewmembers cannot be considered for exchange of rotations in the next two iterations unless aspiration criteria is fulfilled. The following is the most frequently used aspiration criteria (this paper is not an exception): if a swap provides a better solution than the best ever found then swap would be allowed regardless the fact that it is still prohibited according to the recency-based memory. The recency-based memory includes the cells above the main diagonal (Fig. 6). The last two swaps that have been made are (2, 3) and (3, 5). The swap (2, 3) means that crewmember # 2 gave one rotation to crewmember # 3, or that crewmember # 3 gave one rotation to crewmember # 2. Relation (28) helps us to determine the crewmember that should give the rotation. If $z_2 > z_3$ then crewmember # 2 provides rotation, otherwise, crewmember # 3 will give the rotation. Selection of the rotation to be given to the other crewmember is done in the following way: (1) all rotations included in the assignment are sorted in a random manner; (2) by going through the list, an attempt is made to add the rotation to the other crewmember's assignment, until assignment is successful (no constraint is broken) or all rotations in the list are exhausted. In the same way, crewmembers # 3, and # 5 exchange rotations. The frequency-based memory includes the cells below the main diagonal. Each of the cells contains a frequency indicating the number of times that the particular swap has already been made. Thus, for example, we can say that swap (2, 4) has been made 5 times (crewmember # 2, and crewmember # 4 exchanged rotations 5 times in the past), swap (3, 4) four times, swap (1, 5) once, and so on.

Let us denote by F_{ij} the actual value of the objective function (13) obtained after the swap on crewmember pair (i, j) is made. We also denote by F_{ij}^p the penalized value of the objective function. We define this penalized value in the following way:

$$F_{ij}^p = F_{ij}(1 + n_{ij}\Omega), \tag{30}$$

Fig. 6 Recency-based memory and frequency-based memory

	1	2	3	4	5
1					
2	3		2		
3		1			1
4		5	4		
5	1		2	1	

where: n_{ij} the value in the frequency-based memory corresponding to the crewmember pair (i, j), Ω the constant given in advance.

The algorithm for solving the aircrew rostering problem by Tabu Search developed in this paper consists of the following algorithmic steps.

Algorithm 6 (Tabu search algorithm for solving aircrew rostering problem)

Step 1. Randomly create the list of crewmembers (random sort). Using "*pilot by pilot*" method, assign crewmembers to planned rotations. Obtained solution represents current solution. The tenure during which a certain swap is in a tabu status equals τ (value given in advance and kept constant through the search process).

Step 2. Construct the neighborhood of the current solution in the following way. Generate the list of all pairs of crewmembers and perform a swap for each of them. Objective function evaluation is a part of the swap. Using information about tabu status of certain swaps, remove from the list pairs of crewmembers whose swap is tabu and did not lead to the fulfillment of the aspiration criteria. The solutions obtained in this way represent adjacent solutions that can be reached from the current solution.

Step 3. Using expression (30), penalize swaps according to the frequency-based memory. The swaps with greater frequency will receive larger penalty. Sort all solutions in descending order of penalized objective function value. Replace the current solution by the best solution from the sorted list. Update the information related to the tabu status, frequencies and the total number of iterations performed.

Step 4. If the number of performed iterations is less than the number of planned iterations, go to Step 2. Otherwise, end the algorithm.

6 Numerical example

The proposed models were tested on the real numerical example of assigning 53 flight captains to 422 rotations to be executed during a 30-day period. The rotation characteristics are presented in Table 4. Table 5 provides information about pilots' availability during the same 30 days (table is composed of the binary values sequence for each pilot—values p_{il}).

All the results presented in this section are obtained for the following criteria weights: $w_1 = 0.4$, $w_2 = 0.3$, and $w_3 = 0.3$.

Simulated Annealing, Genetic Algorithms and Tabu Search based models developed in this paper are initialized with the following parameters:

(a) The simulated annealing:
1. The total number of temperatures $q = 180$.
2. The initial temperature $t_1 = 1000$.
3. Temperature change parameter is 0.85 ($c = 0.85$). The law on changing temperature reads: $T_i = 0.85T_{i-1}$, $i = 2, 3, \ldots, 180$.

Table 4 Characteristic of the rotations

Rotation number	Rotation starting day	Rotation starting time [min]	The earliest day of the following rotation start	The earliest time of the following rotation start [min]	Number of legs in rotation	Rotation flight time [min]	Rotation working time [min]	Rotation foreign per diem allowance	Rotation working days	Number of weekend days included in rotation
1	1	355	2	270	5	240	485	0	1	0
2	1	365	2	1175	4	355	585	1	2	0
3	1	380	3	340	6	590	1090	2	2	0
4	1	390	2	1105	5	225	445	0	2	0
5	1	400	1	1210	4	210	300	0	1	0
6	1	515	3	790	7	615	1030	2	2	0
–	–	–	–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–	–	–	–
421	30	1260	31	1040	2	105	285	0	2	0
422	30	1325	31	930	2	365	535	2	2	0

Table 5 Pilot availability (p_{il})

Pilot no.	Days							
	1	2	3	4	–	–	29	30
1	1	1	1	1	–	–	1	1
2	1	1	0	1	–	–	1	1
3	1	1	1	1	–	–	1	1
4	1	1	1	1	–	–	0	0
–	–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–	–
52	1	1	1	1	–	–	1	1
53	1	1	1	0	–	–	1	1

Table 6 Summary of the results (SA, GA and TS)

	Flight time				Foreign per diem allowance				Weekend days				CPU
	ave. [%]	stDev [%]	min [%]	max [%]	ave. [%]	stDev [%]	min [%]	max [%]	ave. [%]	stDev [%]	min [%]	max [%]	time [min]
SA	1.06	0.94	0.01	4.30	0	0	0	0	0	0	0	0	20
GA	1.39	2.27	0.02	14.05	0.42	2.14	0	11.11	1.08	3.80	0	14.29	12
TS	2.22	2.66	0.01	10.63	0.80	2.16	0	11.11	2.39	4.58	0	16.67	4

4. The value of parameter ε that figures in the thermal equilibrium expression is $\varepsilon = 0.05$.
5. The maximum number of epochs α at one temperature is $\alpha = 80$.
6. Criteria are raised to the power $r = 3$.
7. Within the scope of perturbation, pilots are changed on five rotations ($n = 5$).
8. The maximum number of attempts to make small changes is 20 ($z = 20$).

(b) Genetic algorithms:
1. The total number of generations is 650.
2. Population size is 140.
3. Crossover probability $p_c = 0.95$.
4. Mutation probability $p_m = 0.005$.

(c) Tabu search:
1. The total number of iterations equals 500.
2. $\Omega = 0.005$(frequency based penalty parameter).
3. The tenure during which a certain swap is in a tabu status equals 13 ($\tau = 13$).

Table 6 provides basic characteristics, such as average (ave.), standard deviation (stDev), minimum (min), and maximum (max), of absolute values of relative deviation of: (1) real flight time from the ideal (the "Flight time" column), (2) real number of foreign per diem allowance from the ideal (the "Foreign per diem allowance" column) and (3) weekend days spent on duty from the ideal (the "Weekend days" column). Furthermore, Table 6 provides required CPU time per run for all three algorithms in minutes. The runs are completed on a PC Pentium-4 2.4 GHz.

Figure 7 depicts the shape of the distribution of relative deviation of real flight time from ideal achieved using SA, GA and TS. Since there are only several instances (flight captains)

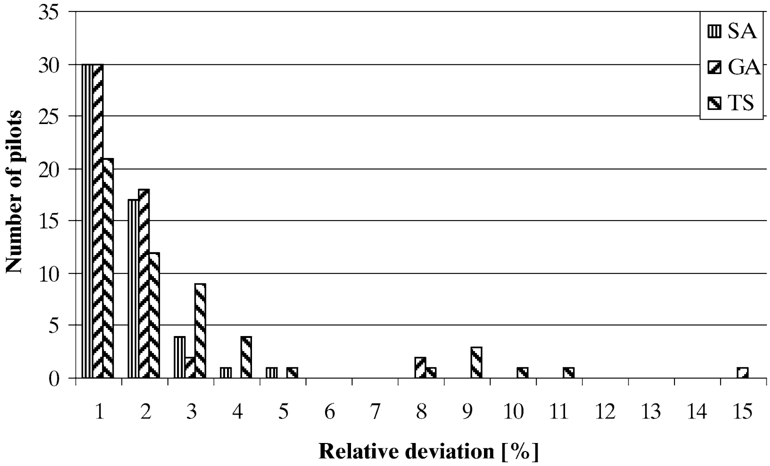


Fig. 7 Distribution of absolute values of relative deviation of real flight time from the ideal—SA, GA and TS

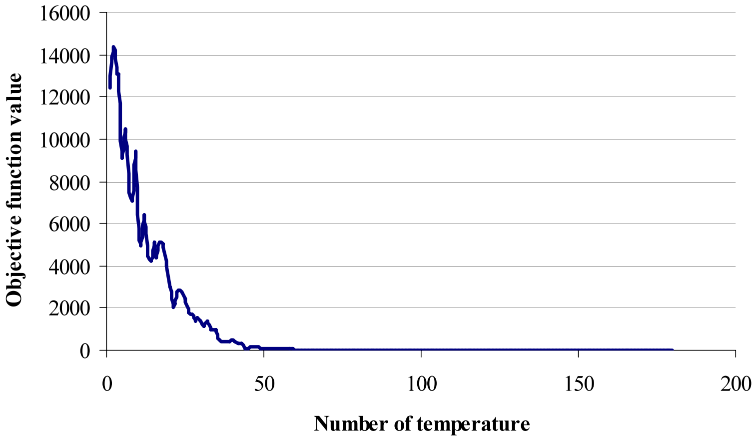


Fig. 8 Change of objective function (F) with temperature change

whose allocation of foreign per diem allowance and weekend days spent on duty differs from ideal distribution of the deviations will not be provided.

Change of objective function F as functions of temperature index is depicted in Fig. 8.

Change of the objective function as a function of generations is depicted in Fig. 9.

Change of the objective function value across iterations of Tabu Search algorithm is depicted in Fig. 10.

Results of testing the models using numerical examples of different dimensions are shown in Table 7.

It is not our intention to make a detailed comparison of Metaheuristics for the Aircrew rostering problem. However, our implementation of metaheuristic algorithms (which may not be the best one) showed that the best results are obtained by the Simulated Annealing technique.

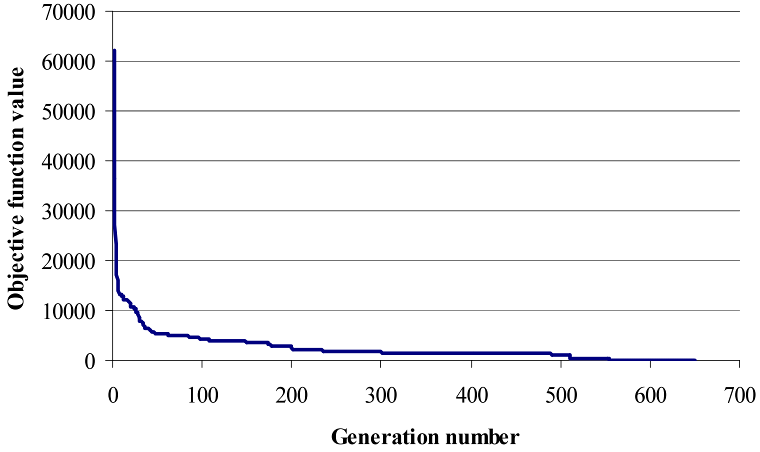


Fig. 9 Change of objective function (F) across generations—genetic algorithms

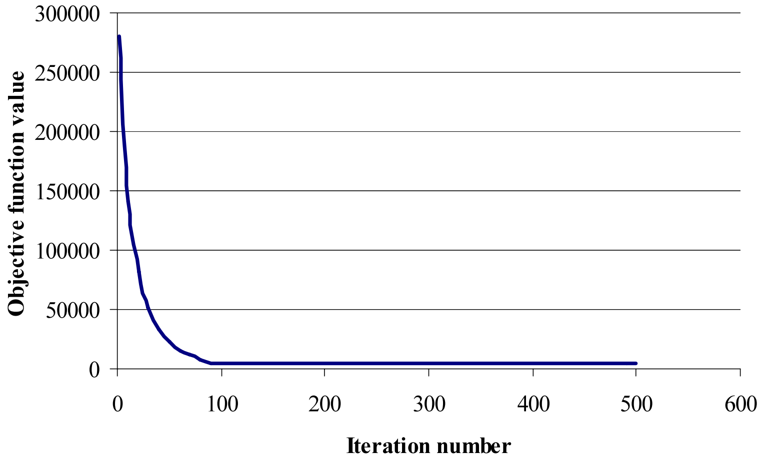


Fig. 10 Change of objective function (F) across iterations—Tabu Search

Table 7 Results of testing the models using examples of different dimensions ($r = 1$)

Number of pilots	Number of rotations	SA			GA			TS		
		$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$
27	221	4.323	0	0	5.816	4.327	2.416	6.113	3.469	5.433
33	285	2.135	0	0	3.695	5.181	1.563	5.720	3.216	6.796
45	352	1.326	0	0	2.450	3.226	1.432	4.030	3.167	4.254
65	580	0.882	0	0	1.293	1.218	0.834	2.023	0.751	2.263

7 Conclusion

This paper offers metaheuristic approaches to solve the aircrew rostering problem. An attempt is made to find appropriate ways to follow the general principles of the metaheuristic algorithms when solving the complex aircrew rostering problem. Taking into account the combinatorial nature of the problem, its possible large dimensions, and the aspiration to solve the problem based on several criteria, the proposed algorithms are heuristic. When solving the crew rostering problem, interests of both the air carrier and the aircrew must be taken into consideration. Clearly, the assignment of pilots to rotations can be made based on objective functions that differ from the objective functions proposed in this paper. The developed algorithms were tested on a real numerical example whose dimensions are characteristic of small and medium-sized airlines. The other examples used to test the proposed algorithms were generated randomly. Since aircrew rostering does not belong to the class of problems that must be solved in real time, the achieved CPU times are satisfactory.

When carrying out a planned airline schedule, certain disturbances arise that the air carrier is not able to foresee. Whenever a disturbance in an airline schedule appears, the dispatcher in charge of traffic management tries to minimize the negative effects resulting from the disturbance (delays, flight cancellations). Mitigation of planned airline schedule disturbances in air transportation network is a typical problem that must be solved in real time. The schedule disturbances, as well as applied strategies to mitigate these disturbances highly influence final aircrew rostering. It is extremely important, in future research of the aircrew scheduling and rostering problems to make an attempt to take into account various aspects of the airline schedule disturbances.

The aircrew rostering problem is a difficult combinatorial optimization problem characterized by multiple goals. An attempt has been made in this paper to consider the multiobjective nature of the aircrew rostering problem (We used a relatively simple objective function $F = w_1 f_1 + w_2 f_2 + w_3 f_3$). In future research of the aircrew rostering problem, more complex models based on Metaheuristic algorithms for multiobjective problems should be developed.

References

Agard, M. (1970). Monthly assignment of stewards by computer. In *AGIFORS symposium proceedings* (Vol. 10).

Anantaram, C., Joshi, P., Deshpande, K., & Trivedi, P. (1993). Crew rostering system—an expert system for scheduling crew for Indian airlines. In *Proceedings of the ninth conference on artificial intelligence for applications*.

Antosik, J. L. (1978). Automatic monthly crew assignment: a new approach. In *AGIFORS symposium proceedings* (Vol. 18).

Buhr, J. (1978.) Four methods for monthly crew assignment: a comparison of efficiency. In *AGIFORS symposium proceedings* (Vol. 18).

Burke, E. K., De Causmaecker, P., Petrović, S., & Vanden Berghe, G. (2001). Fitness evaluation for nurse scheduling problems. In *Proceedings of the 2001 congress on evolutionary computation—CEC'01* (Vol. 2), 1139–1146.

Burke, E. K., De Causmaecker, P., Petrović, S., & Vanden Berghe, G. (2002). A multi criteria meta-heuristic approach to nurse rostering. In *Proceedings of the 2002 congress on evolutionary computation—CEC '02* (Vol. 2), 1197–1202.

Byrne, J. (1988). A preferential bidding system for technical aircrew. In *AGIFORS symposium proceedings* (Vol. 28).

Cappanera, P., & Gallo, G. (2004). A multicommodity flow approach to the crew rostering problem. *Operations Research*, *52*(4), 583–596.

Caprara, A., Toth, P., & Vigo, D. et al. (1998). Modeling and solving the crew rostering problem. *Operations Research*, *46*(6), 820–830.

Cherny, V. (1985). Thermodynamic approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, *45*, 41–51.

Coello Coello, C. A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, *191*, 1245–1287.

Day, P. R., & Ryan, D. M. (1997). Flight attendant rostering for short-haul airline operations. *Operations Research*, *45*, 649–661.

Eglese, R. W. (1990). Simulated annealing: a tool for operational research. *European Journal of Operational Research*, *46*, 271–281.

El Moudani, W., & Mora-Camino, F. (2000). A fuzzy solution approach for the roster planning problem. In *Proceedings of the FUZZ IEEE 2000—the ninth IEEE international conference on fuzzy systems* (Vol. 1).

El Moudani, W., Cosenza, C. A. N., & Mora-Camino, F. (2001). An intelligent approach for solving the airline crew rostering problem. In *Proceedings of the ACS/IEEE international conference on computer systems and applications*.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004a). Staff scheduling and rostering: a review of applications, methods and models. *European Journal of Operational Research*, *153*, 3–27.

Ernst, A. T., Jiang, H., & Krishnamoorthy, M. et al. (2004b). An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, *127*(1–4), 21–144.

Freling, R., Lentink, R. M., & Wagelmans, A. P. M. (2004). A decision support system for crew planning in passenger transportation using a flexible branch-and-price algorithm. *Annals of Operations Research*, *127*(1–4), 203–222.

Gamache, M., & Soumis, F. (1993). *A method for optimally solving the rostering problem*. Cahier du GERAD, G-90-40, Ecole des Hautes Etudes Commerciales, Montreal, Canada.

Giafierri, J., Hamon, J. P., & Lengline, S. (1982). Automatic monthly assignment of medium-haul cabin crew. In *AGIFORS symposium proceedings* (Vol. 22).

Glanert, W. (1984). A timetable approach to the assignment of pilots to rotations. In *AGIFORS symposium proceedings* (Vol. 24).

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*, 433–549.

Glover, F. (1989). Tabu search—Part I. *ORSA Journal of Computing*, *1*, 190–206.

Glover, F. (1990a). Tabu search—Part II. *ORSA Journal of Computing*, *2*, 4–32.

Glover, F. (1990b). Tabu search—a tutorial. *Interfaces*, *20*, 74–94.

Glover, F., & Laguna, M. (1993). Tabu search. In C. R. Reeves (Ed.), *Modern heuristic techniques for combinatorial problems*. Oxford: Blackwell Scientific.

Hoffman, K. L., & Padberg, M. (1993). Solving airline crew scheduling problems by branch-and-cut. *Management Science*, *39*, 657–682.

Kirkpatrick, S., Gellat, L., & Vecchi, M. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680.

Klabjan, D., Johnson, E. L., Nemhauser, G. L., Gelman, E., & Ramaswamy, S. (2001). Airline crew scheduling with regularity. *Transportation Science*, *35*, 359–374.

Klabjan, D., Johnson, E. L., Nemhauser, G. L., Gelman, E., & Ramaswamy, S. (2002). Airline crew scheduling with time windows and plane-count constraints. *Transportation Science*, *36*, 337–348.

Kohl, N., & Karisch, S. E. (2004). Airline crew rostering: problem types, modeling, and optimization. *Annals of Operations Research*, *127*(1-4), 223–257.

Lučić, P., & Teodorović, D. (1999). Multi-objective simulated annealing model for aircrew rostering problem. *Transportation Research*, *33A*, 19–45.

Marchettini, F. (1980). Automatic monthly cabin crew rostering procedure. In *AGIFORS symposium proceedings* (Vol. 20).

Metropolis, N., Rosenbluth, A., Rosenbluth, M., & Teller, M. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, *21*, 1087–1092.

Moore, R., Evans, J., & Noo, H. (1978). Computerized tailored blocking. In *AGIFORS symposium proceedings* (Vol. 18).

Nicoletti, B. (1975). Automatic crew rostering. *Transportation Science*, *9*, 33–42.

Ryan, D. M. (1992). The solution of massive generalized set partitioning problems in air crew rostering. *Journal of the Operational Research Society*, *43*, 459–467.

Ryan, D. M. (2000). The solution of massive generalized set partitioning problems in aircrew rostering. *Journal of The Operational Research Society*, *51*, 459–467.

Sarra, D. (1988). The automatic assignment model. In *AGIFORS symposium proceedings* (Vol. 28).

Serafini, P. (1994). Simulated annealing for multi objective optimization problems. In G.H.

Teodorović, D. (1988). *Airline operations research*. London: Gordon and Breach.

Teodorović, D., & Lučić, P. (1998). A fuzzy set theory approach to the aircrew rostering problem . *Fuzzy Sets and Systems*, *95*, 261–271.

Tingley, G. A. (1979). Still another solution method for the monthly aircrew assignment problem. In *AGIFORS symposium proceedings* (Vol. 19).

Triki, E., Collette, Y., & Siarry, P. (2005). A theoretical study on the behavior of simulated annealing leading to a new cooling schedule. *European Journal of Operational Research*, *166*, 77–92.

Vance, P. H., Barnhart, C., Johnson, E. L., & Nemhauser, J. L. (1997). Airline crew scheduling: a new formulation and decomposition algorithm. *Operations Research*, *45*, 188–200.

Yan, S., Tung, T.-T., & Tu, Y.-P. (2002). Optimal construction of airline individual crew pairings. *Computers & Operations Research*, *29*(4), 341–363.